A Process-Oriented Heuristic for Model Selection

Pedro Domingos
Artificial Intelligence Group
Instituto Superior Técnico
Lisbon 1096, Portugal
pedrod@gia.ist.utl.pt

Abstract

Current methods to avoid overfitting are either data-oriented (using separate data for validation) or representation-oriented (penalizing complexity in the model). This paper proposes process-oriented evaluation, where a model's expected generalization error is computed as a function of the search process that led to it. The paper develops the necessary theoretical framework, and applies it to one type of learning: rule induction. A process-oriented version of the CN2 rule learner is empirically compared with the default CN2. The process-oriented version is more accurate in a large majority of the datasets, with high significance, and also produces simpler models. Experiments in artificial domains suggest that process-oriented evaluation is particularly useful in high-dimensional domains.

1 INTRODUCTION

Overfitting avoidance is often considered the central problem of machine learning (e.g., (Cheeseman & Oldford, 1994)). If a learner is sufficiently powerful, it must guard against selecting a model that fits the training data well but captures the underlying phenomenon poorly. Current methods to address this problem fall into two broad categories. *Data-oriented evaluation* uses separate data to learn and validate models, and includes methods like cross-validation (Breiman, Friedman, Olshen & Stone, 1984; Stone, 1974), the bootstrap (Efron & Tibshirani, 1993), and reduced-error pruning (Brunk & Pazzani, 1991). It has several disadvantages: it is often computationally intensive, reduces the data available for learning, can be unreliable if the validation set is small, and is itself prone to overfitting if a large number of models is compared (Ng, 1997). *Representation-oriented evaluation* seeks to avoid these problems by using the same data for training and validation, but *a priori* penalizing some models as more likely to overfit. Bayesian approaches in general fall into this category (Cheeseman, 1990; MacKay, 1992). Representation-oriented measures typically contain two terms, one reflecting fit to the data, and one penalizing model complexity (Akaike, 1978; Schwarz, 1978; Wallace & Boulton, 1968; Rissanen, 1978; Moody, 1992). This approach is only appropriate when the simpler models are truly the more accurate ones, and there is mounting evidence that this is typically not the case ((Domingos, 1998; Domingos, 1997; Schuurmans, Ungar & Foster, 1997; Lawrence, Giles & Tsoi, 1997; Webb, 1996; Schaffer, 1993; Murphy & Pazzani, 1994), etc.). Structural risk minimization (Vapnik, 1995) and PAC learning (Kearns & Vazirani, 1994) are representation-oriented methods that seek to bound the difference between training and generalization error using a function of the model space's (effective) dimension. This typically produces bounds that are overly broad, and requires severely restricting the model space.

In this paper we argue that representation-oriented evaluation has these limitations because it only considers the learner's model space, and not its search process. A learner with an unlimited model space can avoid overfitting as long as it attempts only a limited number of hypotheses (even if it is not possible *a priori* to predict which). If these hypotheses are correlated, the chance of overfitting is further reduced. Given the sequence of hypotheses that a learner attempts, it is possible to estimate the generalization error of the "current best" hypothesis taking into account the process that led to it. Intuitively, the more hypotheses

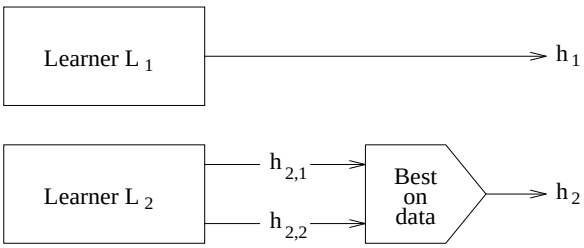

Figure 1: A simple example of an overfitting avoidance problem.

that have been attempted and the less correlated they are, the higher the generalization error we expect for a given training-set error. This paper begins to develop this approach, which we will call *process-oriented evaluation* (POE for short). The basic theoretical framework is presented, and then applied to the standard "separate and conquer" rule induction process (Clark & Niblett, 1989). An empirical study demonstrates the effectiveness of POE. The paper concludes with sections on related and future work.

2 PROCESS-ORIENTED EVALUATION

Consider the simplest example of an overfitting avoidance problem, in a classification context. Suppose learner L_1 consists of drawing one hypothesis at random from some model space and returning it, and learner L_2 consists of drawing two hypotheses at random (independently) from the same model space as L_1, and returning the one with lowest error on a training sample S. This situation is shown schematically in Figure 1. Let h_1 be the hypothesis returned by L_1, h_2 the hypothesis returned by L_2, n the number of examples in S, and e_i the number of examples h_i misclassifies. The goal is to choose the hypothesis with lowest true error ϵ_i (i.e., ϵ_i is the probability of h_i misclassifying an example, given the true example distribution). Suppose $n = 100$, $e_1 = 12$, and $e_2 = 11$. Should we prefer h_1 or h_2? According to the maximum likelihood principle (DeGroot, 1986), $\hat{\epsilon}_1 = 0.12$ and $\hat{\epsilon}_2 = 0.11$, so h_2 should be chosen. Assuming the two hypotheses have the same complexity or prior probability, representation-oriented evaluation would give the same answer. However, L_2 had two opportunities to draw a hypothesis with low training error, and so the probability of e_2 being low merely by chance is higher than for e_1. Thus h_2 may in fact have a higher true error rate than h_1.

This notion can be quantified. If a hypothesis h's true error rate is ϵ and S consists of n independently drawn examples, the number of errors e committed by h on S is a binomially distributed variable with parameters n and ϵ:

$$p(e|n,\epsilon) = b(e|n,\epsilon) = \left(\begin{array}{c} n \\ e \end{array} \right) \epsilon^e (1-\epsilon)^{n-e} \qquad (1)$$

Let $B(e|n,\epsilon)$ be the probability that the number of errors is greater than e:

$$B(e|n,\epsilon) = \sum_{i=e+1}^{n} b(e|n,\epsilon) \qquad (2)$$

Notice that this notation is the opposite of the usual notation for a cumulative distribution function (i.e., $B(e|n,\epsilon) = 1 - \text{Binomial_cdf}(e|n,\epsilon)$). It will be more convenient for what follows.

The probability of h_1 misclassifying e_1 examples is $p(e_1|n,\epsilon_1) = b(e_1|n,\epsilon_1)$. This can be used with Bayes's theorem to compute the expected value of ϵ_1 given n and e_1, $E[\epsilon_1|n,e_1]$. By finding a similar expression for $p(e_2|n,\epsilon_2)$, we can compute $E[\epsilon_2|n,e_2]$ and choose the hypothesis with lowest expected error. Let the two hypotheses drawn by L_2 be $h_{2,1}$ and $h_{2,2}$ (with true errors $\epsilon_{2,1}$ and $\epsilon_{2,2}$ respectively, and numbers of training errors $e_{2,1}$ and $e_{2,2}$). From these, L_2 chooses the one with lowest training error (i.e., $h_2 = h_{2,j}$, where $j = \text{argmin}_{i \in 1,2}\, e_{2,i}$). Then the probability of L_2 returning a hypothesis h_2 that misclassifies e_2 training examples is the probability that $h_{2,1}$ misclassifies e_2 training examples and $h_{2,2}$ misclassifies more, or vice-versa, or both $h_{2,1}$ and $h_{2,2}$ misclassify e_2 examples:

$$\begin{aligned} p(e_2|n,\epsilon_2) \;=\; & b(e_2|n,\epsilon_{2,1})B(e_2|n,\epsilon_{2,2}) \\ & + B(e_2|n,\epsilon_{2,1})b(e_2|n,\epsilon_{2,2}) \\ & + b(e_2|n,\epsilon_{2,1})b(e_2|n,\epsilon_{2,2}) \end{aligned} \qquad (3)$$

Our goal is to use this equation to compute the expected value of ϵ_2. We are hindered by the fact that in addition to ϵ_2 (whether it is $\epsilon_{2,1}$ or $\epsilon_{2,2}$) the equation contains another unknown parameter (whichever $\epsilon_{2,i}$ is not ϵ_2). Since we are not interested in $\epsilon_{2,1}$ or $\epsilon_{2,2}$ *per se*, but only in the effect on ϵ_2 of trying two hypotheses instead of of one, we propose the following heuristic: assume that $\epsilon_{2,1} = \epsilon_{2,2} = \epsilon_2$. This approximation will be good if $\epsilon_{2,1}$ and $\epsilon_{2,2}$ are similar, and poor if they are very different. However, this heuristic

may yield good results even in the latter case, because a close approximation of $E[\epsilon_2|n,e_2]$ is not required; all that is required is that $E[\epsilon_2|n,e_2] > E[\epsilon_1|n,e_1]$ iff $\epsilon_2 > \epsilon_1$, which is a much weaker condition (Domingos & Pazzani, 1997). If $\epsilon_{2,1} = \epsilon_{2,2} = \epsilon_2$ Equation 3 becomes:

$$\begin{aligned} p(e_2|n,\epsilon_2) &= b(e_2|n,\epsilon_2)B(e_2|n,\epsilon_2) \\ &\quad + B(e_2|n,\epsilon_2)b(e_2|n,\epsilon_2) \\ &\quad + b(e_2|n,\epsilon_2)b(e_2|n,\epsilon_2) \\ &= [B(e_2|n,\epsilon_2) + b(e_2|n,\epsilon_2)]^2 \\ &\quad - B^2(e_2|n,\epsilon_2) \\ &= B^2(e_2-1|n,\epsilon_2) - B^2(e_2|n,\epsilon_2) \end{aligned} \quad (4)$$

Applying Bayes's theorem:

$$p(\epsilon_2|n,e_2) \propto p(\epsilon_2)p(e_2|n,\epsilon_2) \quad (5)$$

$p(\epsilon_2)$ can be used to incorporate prior beliefs about the error rate of the hypotheses considered by L_2. Here it will simply be assumed uniform:[1]

$$p(\epsilon_2|n,e_2) \propto p(e_2|n,\epsilon_2) \quad (6)$$

The expected value of ϵ_2 can now be computed by integration:

$$E[\epsilon_2|n,e_2] = \frac{\int_0^1 \epsilon_2\, p(e_2|n,\epsilon_2)\, d\epsilon_2}{\int_0^1 p(e_2|n,\epsilon_2)\, d\epsilon_2} \quad (7)$$

Doing this for $e_2 = 11, n = 100$ results in $E[\epsilon_2|n,e_2] = 0.134$. A similar treatment for e_1, using $e_1 = 12$, $n = 100$ and $p(e_1|n,\epsilon_1) = b(e_1|n,\epsilon_1)$, yields $E[\epsilon_1|n,e_1] = 0.127$. Thus the hypothesis output by L_1 would be preferred, even though L_2's has a lower training error.

Equation 4 can be readily generalized to a learner L_m that draws m hypotheses at random and chooses the one with lowest training error:

$$\begin{aligned} p(\epsilon_m|n,e_m) &\propto p(e_m|n,\epsilon_m) \\ &= B^m(e_m-1|n,\epsilon_m) - B^m(e_m|n,\epsilon_m) \end{aligned} \quad (8)$$

Notice that this formula makes intuitive sense: as m increases, the mass of probability is shifted to higher and higher ϵ_m's; but as n increases, higher and higher m's are needed to make this happen to the same degree. To see this, consider the binomial expansion

$$\begin{aligned} &B^m(e_m-1|n,\epsilon_m) \\ &= [B(e_m|n,\epsilon_m) + b(e_m|n,\epsilon_m)]^m \\ &= B^m(e_m|n,\epsilon_m) + mB^{m-1}(e_m|n,\epsilon_m)b(e_m|n,\epsilon_m) \\ &\quad + \frac{m(m-1)}{2}B^{m-2}(e_m|n,\epsilon_m)b^2(e_m|n,\epsilon_m) + \cdots \end{aligned} \quad (9)$$

and consider that, for all but the smallest sample sizes, $B(e_m|n,\epsilon_m) \gg b(e_m|n,\epsilon_m)$. Thus:

$$\begin{aligned} p(\epsilon_m|n,e_m) &\propto p(e_m|n,\epsilon_m) \\ &= B^m(e_m-1|n,\epsilon_m) - B^m(e_m|n,\epsilon_m) \\ &\simeq mb(e_m|n,\epsilon_m)B^{m-1}(e_m|n,\epsilon_m) \end{aligned} \quad (10)$$

When $m = 1$, this reduces to $b(e_m|n,\epsilon_m)$, as expected. When $m = 2$, $b(e_m|n,\epsilon_m)$ is multiplied by a constant and by $B(e_m|n,\epsilon_m)$. Since the latter is a function that increases monotonically with ϵ_m for a given n and e_m, the effect of this is to decrease the probability of lower ϵ_m's and increase the probability of higher ones, and thus to increase the expected ϵ_m. As m increases, $b(e_m|n,\epsilon_m)$ is multiplied by higher and higher powers of $B(e_m|n,\epsilon_m)$. This further decreases the probability of low ϵ_m's and increases the probability of high ones, leading to an ever-increasing expected ϵ_m. As an example, Figure 2 shows $b(25|50,\epsilon_m)$ (magnified by a factor of five) and several powers of $B(25|50,\epsilon_m)$. The resulting $E[\epsilon_m|50,25]$ (not shown) has a roughly similar shape to $b(25|50,\epsilon_m)$, but shifts rightward in step with $B(25|50,\epsilon_m)$. For larger n, the same process takes place, but $b(e_m|n,\epsilon_m)$ is more sharply peaked, $B(e_m|n,\epsilon_m)$ also transitions from values close to zero to values close to one more sharply, and the advance of $B^m(e_m|n,\epsilon_m)$ to the right becomes correspondingly slower (since, for any $0 < k < y < 1$, as $y \to 1$ with

[1] This is an unrealistic assumption, and is made solely for the sake of simplicity. As the following sections show, the proposed method can be effective even when this assumption is used. This can be attributed to the fact that, except for very small sample sizes and/or very extreme priors, the effect of the likelihood term $p(e_2|n,\epsilon_2)$ will easily dominate the prior's. In any case, a version of POE using beta priors is currently being implemented.

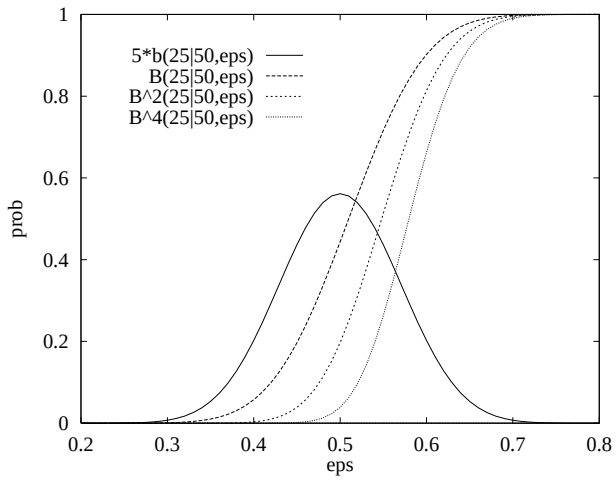

Figure 2: Variation of $b(e_m|n,\epsilon_m)$ and powers of $B(e_m|n,\epsilon_m)$ with ϵ_m for $n = 50, e_m = n/2$.

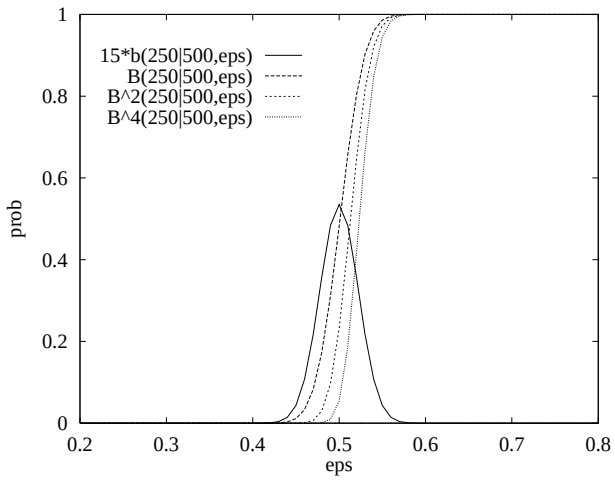

Figure 3: Variation of $b(e_m|n,\epsilon_m)$ and powers of $B(e_m|n,\epsilon_m)$ with ϵ_m for $n = 500, e_m = n/2$.

k held constant higher and higher m's are needed to make $y^m \leq k$). This can be seen by comparing Figure 2 with Figure 3, which shows the corresponding plots for $n = 500$.

Equation 8 still assumes that all m hypotheses drawn are independent, but it can be further generalized to include the dependent case:

$$\begin{aligned} p(\epsilon_m|n,e_m) &\propto p(e_m|n,\epsilon_m) \\ &= p(\forall_{1\leq i\leq m}\, e_{m,i} \geq e_m|n,\epsilon_m) \\ &\quad -p(\forall_{1\leq i\leq m}\, e_{m,i} > e_m|n,\epsilon_m) \end{aligned} \quad (11)$$

Evaluating this expression when high-order dependencies are present will generally not be feasible, but the standard Bayesian network approach (Heckerman, 1996) is applicable here: the number of training errors $e_{m,i}$ of each hypothesis $h_{m,i}$ generated by L_m can be viewed as a node in a Bayesian network, whose parents are the training errors of the hypotheses $h_{m,j}$ it is primarily dependent on. For example, in many greedy search processes (e.g., standard decision tree induction), if $h_{m,3}$ was derived from $h_{m,2}$, which in turn was derived from $h_{m,1}$, $e_{m,3}$ will be approximately independent of $e_{m,1}$ given $e_{m,2}$. In general, the Bayesian network for a given learning process will have the DAG (directed acyclic graph) of the search process itself as a subgraph (e.g., in a greedy search each node $e_{m,i}$ will have arcs to the training errors of the hypotheses that were generated from $h_{m,i}$). If $par(e_{m,i})$ are the parents of $e_{m,i}$ in the Bayesian network, Equation 11 above reduces to:

$$\begin{aligned} p(\epsilon_m|n,e_m) \propto p(e_m|n,\epsilon_m) = \\ \prod_{i=1}^{m} p(e_{m,i} \geq e_m|n,\epsilon_m,\forall_{e_{m,j}\in par(e_{m,i})}\, e_{m,j} \geq e_m) \\ -\prod_{i=1}^{m} p(e_{m,i} > e_m|n,\epsilon_m,\forall_{e_{m,j}\in par(e_{m,i})}\, e_{m,j} > e_m) \end{aligned} \quad (12)$$

L_1 and L_2 above were considered to be different learners, but they can equally well be considered different stages of the same learner. For example, L_2 can take the hypothesis output by L_1 as its own first hypothesis. More generally, L_m can be the result of continuing the search of learner L_k ($k < m$) with $m-k$ more hypotheses. Thus this framework can be applied to problems like decision tree and rule pruning, to which we now turn.

3 AN APPLICATION: RULE INDUCTION

Most rule induction systems employ a set covering or "separate and conquer" search strategy (Michalski, 1983; Clark & Niblett, 1989). Rules are induced one at a time, and each rule starts with a training set composed of the examples not covered by any previous rules. A rule is induced by adding conditions one at a time, starting with none (i.e., the rule initially covers the entire instance space). The next condition to add is chosen by attempting all possible conditions. Con-

ditions on symbolic attributes are typically of the form $a_i = v_{ij}$, where v_{ij} is a possible value of attribute a_i. Conditions on numeric attributes are typically of the form $a_i \leq v_{ij}$ or $a_i > v_{ij}$, where the thresholds v_{ij} are usually values of the attribute that appear in the training set. In the beam search process used by many rule learners, at each step the best b versions of the rule according to some evaluation function are selected for further specialization. AQ (Michalski, 1983) continues adding conditions until the rule is "pure" (i.e., until it covers examples of only one class). This can lead to severe overfitting. The latest version of the CN2 system (Clark & Boswell, 1991) uses a simple and effective Bayesian method to combat this: induction of a rule stops when no specialization improves its error rate, and the latter is computed using a *Laplace correction* or m-*estimate*. If n_r is the number of examples covered by a rule r, and e_r is the number of those examples it misclassifies, the conventional estimate of the rule's error rate is e_r/n_r, but its m-estimate is:

$$\hat{\epsilon}_r = \frac{e_r + \mathsf{m}\epsilon_0}{n_r + \mathsf{m}} \tag{13}$$

where ϵ_0 is the rule's *a priori* error, which CN2 takes to be the error obtained by random guessing if all classes are equally likely: $\epsilon_0 = (c-1)/c$, where c is the number of classes. This prior value is given a weight of m examples (i.e., the behavior of Equation 13 is equivalent to having m additional examples covered by the rule, one of each class). CN2 uses m=c. As conditions are added, the rule covers fewer and fewer examples, and $\hat{\epsilon}_r$ tends to ϵ_0. Thus a rule making more misclassifications may be preferred if it covers more examples, causing induction to stop earlier and reducing overfitting. Clark and Boswell (Clark & Boswell, 1991) found this version of CN2 to be more accurate than C4.5 (Quinlan, 1993) on 10 of the 12 benchmark datasets they used for testing. However, this scheme ignores that, as more and more conditions are attempted, the probability of finding one that appears to reduce the rule's error merely by chance increases. This will lead the m-estimate to underestimate the chosen condition's true error, and CN2 to overfit. The upward correction made to ϵ_r should increase with the number of conditions attempted. The process-oriented evaluation framework described in the previous section allows us to do this in a systematic way.

Let each hypothesis be one version of the rule attempted during the beam search. The main change to Equation 8 required is to take into account that different versions of a rule will cover different numbers of training examples. In other words, n is now a function of the hypothesis, and the hypothesis with lowest e_i/n_i is chosen. Let $\vec{n}_m = (n_1, \ldots, n_i, \ldots, n_m)$, where n_i is the number of examples covered by rule version i, and let $\hat{\epsilon}_m = \min_{1 \leq i \leq m} \{e_i/n_i\}$ be the lowest training-set error rate found so far. Equation 8 becomes:

$$p(\epsilon_m | \vec{n}_m, \hat{\epsilon}_m) \propto p(\hat{\epsilon}_m | \vec{n}_m, \epsilon_m) = \prod_{i=1}^{m} B(n_i\hat{\epsilon}_m - 1 | n_i, \epsilon_m) - \prod_{i=1}^{m} B(n_i\hat{\epsilon}_m | n_i, \epsilon_m) \tag{14}$$

This equation does not need to be computed for every rule version generated during the beam search, but only once for each round. One round consists of generating every possible one-step specialization of each rule version in the beam, and selecting the b best. Thus, if there are a attributes and v is the maximum number of values of any attribute (in the worst case, $v = n$ for numeric attributes), one round corresponds to $O(bav)$ rule versions. Let m_k be the total number of rule versions generated up to, and including, round k. Round 1 consists of the initial rule with no conditions, and $m_1 = 1$. Induction stops when $E[\epsilon_{m_k} | \vec{n}_{m_k}, \hat{\epsilon}_{m_k}] \geq E[\epsilon_{m_{k-1}} | \vec{n}_{m_{k-1}}, \hat{\epsilon}_{m_{k-1}}]$, for $k > 1$.

Equation 14 is of course only a first approximation. Many other aspects of the rule induction process can be taken into account using Equation 12, and making approximations as needed for computational efficiency. A version of CN2 that takes into account the dependence between each rule version and its parent (i.e., the rule version it specializes by one condition) is currently being implemented.

4 EMPIRICAL STUDY

In order to test the effectiveness of process-oriented evaluation, default and process-oriented versions of CN2 were compared on the benchmark datasets previously used by Clark and Boswell (1991).[2] The process-oriented version was implemented by adding the necessary facilities to the CN2 source code. Numerical integration (Equation 7) was performed using Simpson's rule, and $B(e|n,\epsilon)$ (Equation 2) was computed using the incomplete beta function (Press, Teukolsky, Vetterling & Flannery, 1992). Integrating Equation 14 every time $E[\epsilon_{m_k} | \vec{n}_{m_k}, \hat{\epsilon}_{m_k}]$ needs to be computed (once

[2] With the exception of pole-and-cart, which is not available in the UCI repository (Merz, Murphy & Aha, 1997).

per round) would generally significantly slow down the rule induction process. Instead, it was approximated by:

$$p(\epsilon_m|\bar{n},\hat{\epsilon}_m) \propto p(\hat{\epsilon}_m|\bar{n},\epsilon_m) = B^m(\bar{n}\hat{\epsilon}_m - 1|\bar{n},\epsilon_m) - B^m(\bar{n}\hat{\epsilon}_m|\bar{n},\epsilon_m) \quad (15)$$

where $\bar{n} = \frac{1}{m}\sum_{i=1}^{m} n_i$. This replaces each of the products with a single-step computation, speeding up evaluation by $O(m)$. CN2's Laplace estimates are still used to choose the best b specializations in each round. This is preferable to using uncorrected estimates, since as implemented POE has no preference between hypotheses within the same round, and this is also a factor in avoiding overfitting. However, the Laplace correction distorts the values used by Equation 15. This will be particularly pronounced when there are many classes, since CN2 uses $\mathsf{m} = c$. In order to minimize this problem, $\mathsf{m} = 2$ was used with POE.[3]

The experimental procedure of (Clark & Boswell, 1991) was followed. Each dataset was randomly divided into 67% for training and 33% for testing, and the error rate and theory size (total number of conditions) were measured for default CN2 and CN2-POE. This was repeated 20 times. The average results and their standard deviations are shown in Table 1.[4]

POE reduces CN2's error rate in 8 of the 11 datasets. Using a sign test, these results are significant at the 4% level. In other words, POE improves CN2 with high confidence. It also produces simpler rule sets in all but two of the datasets. With the approximation used, POE did not noticeably increase CN2's running time. This is also due to the fact that POE tends to make induction stop sooner than in default CN2, as evinced by the theory size results.

While these results are encouraging, they do not necessarily prove that CN2-POE reduces overfitting by taking into account the increasing number of rule versions generated as search progresses. If this is indeed what is taking place, the difference in error between default CN2 and CN2-POE ($error_{CN2} - error_{CN2-POE}$) should increase with the dataset's number of attributes, since this will increase the number of rule versions generated in each round. In order to test this hypothesis, experiments were carried out in artificial domains. Concepts defined as Boolean functions in disjunctive normal form were used as targets. The datasets were composed of 100 training examples and 1000 test examples described by a variable number of attributes a. The number of literals d in each disjunct was generated at random, with a mean of $d = 5$ and a variance of $5 \times (1 - \frac{5}{a})$. This is obtained by including each literal in the disjunct with probability $\frac{5}{a}$. Literals were negated or not with equal probability. The number of disjuncts was set to $2^{d-1} = 16$, which ensures the concept covers roughly half the instance space. Equal numbers of positive and negative examples were included in the dataset, and positive examples were divided evenly among disjuncts. In each run a different target concept was used. One hundred runs were conducted for each value of a between 10 and 100 (at intervals of 5), and the correlation between ($error_{CN2} - error_{CN2-POE}$) and a was measured. This was found to be highly positive ($\rho = 0.66$), confirming our hypothesis.

5 RELATED WORK

The literature on model selection and error estimation is very large, and we will not attempt to review it here. The incompleteness of representation-oriented evaluation was noted 20 years ago by Pearl (1978):

> It would, therefore, be more appropriate to connect credibility with the nature of the selection procedure rather than with properties of the final product. When the former is not explicitly known ... simplicity merely serves as a rough indicator for the type of processing that took place prior to discovery.

Huber (St. Amant & Cohen, 1997; Huber, 1994) expresses thus the need for process-oriented evaluation:

> Data analysis is different from, for example, word processing and batch programming: the correctness of the end product cannot be checked without inspecting the path leading to it.

Several pieces of previous work take into account the number of hypotheses being compared, and so can be considered early steps towards process-oriented evaluation. This includes notably systems that use the

[3] Simply changing $\mathsf{m} = c$ to $\mathsf{m} = 2$ in default CN2 does not change its performance on the datasets used.

[4] There are some differences between CN2's results and those reported in (Clark & Boswell, 1991). This may be due to the fact that the default version of CN2 uses a beam size of 5, whereas Clark and Boswell used $b = 20$. The distribution version of CN2 may also differ from the one used in (Clark & Boswell, 1991).

Table 1: Empirical results: error rates and theory sizes of default CN2 and CN2 with process-oriented evaluation (CN2-POE).

Dataset	Error rate		Theory size	
	CN2	CN2-POE	CN2	CN2-POE
Breast	30.0±1.4	29.7±1.4	114.5±2.4	58.7±2.6
Echocardio	32.7±1.2	32.3±1.3	42.9±1.2	35.4±2.1
Glass	39.0±1.5	38.3±1.7	51.8±1.0	54.7±1.1
HeartC	20.8±0.8	22.5±0.8	57.8±0.9	52.0±1.0
HeartH	22.4±1.1	21.8±1.3	69.2±1.5	60.3±1.4
Hepatitis	21.2±0.9	19.2±1.3	40.2±1.7	34.0±1.3
Lympho	21.4±1.1	24.1±1.1	39.5±0.7	38.7±1.0
Soybean	19.5±1.0	19.4±1.0	116.7±2.3	110.9±3.1
Thyroid	4.1±0.2	3.8±0.2	97.5±2.0	104.8±2.0
Tumor	60.1±1.0	65.1±1.3	302.8±4.6	273.9±4.4
Voting	4.8±0.4	4.3±0.3	61.7±2.9	49.6±2.5

Bonferroni correction when testing significance (e.g., (Kass, 1980; Gaines, 1989; Jensen & Schmill, 1997); see also (Miller, 1981; Klockars & Sax, 1986; Westfall & Wolfinger, 1997)). A key difference between these systems and what is proposed here is that they require a somewhat arbitrary choice of significance threshold, while this paper directly attempts to optimize the end goal (expected generalization error). Also, the Bonferroni correction does not take hypothesis dependencies into account, while the present framework offers (at least in principle) a way of doing so.

Quinlan and Cameron-Jones's (1995) "layered search" method for automatically selecting CN2's beam width can also be considered a form of process-oriented evaluation. While layered search and CN2-POE have similar aims, their biases differ: layered search limits the search's width, while CN2-POE limits its length. The latter may be more effective in reducing the fragmentation and small disjuncts problems (Pagallo & Haussler, 1990; Holte, Acker & Porter, 1989). The assumptions made by the heuristic proposed here are also clearer than those implicit in Quinlan and Cameron-Jones's measure.

Evaluating models that are the result of a search process, not just of fitting the parameters of a predetermined structure, has traditionally not been a concern of statisticians. However, this is beginning to change (Chatfield, 1995).

Some of the arguments made here for taking into account the number of hypotheses attempted are made in greater detail in (Cohen & Jensen, 1997) and (Ng, 1997). The present paper goes further in arguing that other aspects of the search process should also be taken into account whenever possible (for example, in rule induction, the number of examples covered by each hypothesis).

6 FUTURE WORK

The development and evaluation contained in this paper are obviously only preliminary. As mentioned above, a version of CN2-POE that takes hypothesis dependencies into account is currently being implemented. Applications of POE to decision tree induction, backpropagation, instance selection, feature selection and discretization are also areas for future work. In each case, the main issue is likely to be finding the optimal trade-off between the computational and mathematical complexity of POE and its payoff in reduced error rates. The success of the enterprise is likely to hinge on distinguishing strong dependencies from weak ones that can be ignored, and on finding efficient but roughly correct approximations. For most learners in most domains, it is probably not realistic to expect large error reductions from POE, since it does not change the underlying representation or search process. However, if POE's gains are small but consistent across a broad spectrum of learners and domains, it will still be worth developing.

The POE error estimates introduced in this paper have two types of statistical bias. One stems from the fact that, because evaluation focuses on the lowest error found, low outliers have a stronger effect than high ones, leading to a negative bias (i.e., underestimating error). This bias can be estimated and the POE values corrected. This is an area of current work. The

second source of bias is the assumption that all hypotheses tried by the learner have similar error rates. This will lead to a positive bias when the error rate is decreasing (i.e., POE will tend to overestimate error at least up to the point where the learner starts overfitting). One way to overcome this is to introduce explicit expectations about the evolution of the learner's error as search progresses. For example, a specific type of curve may be assumed, or an "expected curve" can be compiled by cross-validation. Another approach is to avoid the assumption of similar error rates, for example by marginalizing over the true error rates of all hypotheses but the chosen one, or by using their maximum-likelihood estimates. Both of these approaches are also currently being studied.

The ultimate goal of POE is to accurately predict a hypothesis's generalization error from its training-set error, using knowledge of how the hypothesis was obtained. How far this is possible remains an open question.

7 CONCLUSION

Two main types of model selection are currently available. In *data-oriented evaluation*, a hypothesis's score does not depend on its form or how the hypothesis was found, but only on its performance on the data. In *representation-oriented evaluation*, the score depends on the data and on the hypothesis's form, but not on the search process that led to it. This paper argued that the latter cannot be ignored, and proposed *process-oriented evaluation* (POE), which takes all three factors into account. An application of POE to the CN2 rule induction system was found to reduce error in 8 of 11 benchmark datasets, and produce simpler theories in 9. Experiments in artificial domains support the hypothesis that these gains stem at least partly from CN2-POE's use of search process information.

References

Akaike, H. (1978). A Bayesian analysis of the minimum AIC procedure. *Annals of the Institute of Statistical Mathematics, 30A*, 9–14.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees.* Belmont, CA: Wadsworth.

Brunk, C., & Pazzani, M. J. (1991). An investigation of noise-tolerant relational concept learning algorithms. *Proceedings of the Eighth International Workshop on Machine Learning* (pp. 389–393). Evanston, IL: Morgan Kaufmann.

Chatfield, C. (1995). Model uncertainty, data mining and statistical inference. *Journal of the Royal Statistical Society A, 158.*

Cheeseman, P. (1990). On finding the most probable model. In J. Shrager & P. Langley (Eds.), *Computational Models of Scientific Discovery and Theory Formation* (pp. 73–95). San Mateo, CA: Morgan Kaufmann.

Cheeseman, P., & Oldford, R. W. (1994). Preface. In P. Cheeseman & R. W. Oldford (Eds.), *Selecting Models from Data: Artificial Intelligence and Statistics IV.* New York: Springer-Verlag.

Clark, P., & Boswell, R. (1991). Rule induction with CN2: Some recent improvements. *Proceedings of the Sixth European Working Session on Learning* (pp. 151–163). Porto, Portugal: Springer-Verlag.

Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning, 3*, 261–283.

Cohen, P. R., & Jensen, D. (1997). Overfitting explained. *Preliminary Papers of the Sixth International Workshop on Artificial Intelligence and Statistics* (pp. 115–122). Fort Lauderdale, FL: Society for Artificial Intelligence and Statistics.

DeGroot, M. H. (1986). *Probability and Statistics* (2nd ed.). Reading, MA: Addison-Wesley.

Domingos, P. (1997). Why does bagging work? A Bayesian account and its implications. *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining* (pp. 155–158). Newport Beach, CA: AAAI Press.

Domingos, P. (1998). Occam's two razors: The sharp and the blunt. Submitted.

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning, 29*, 103–130.

Efron, B., & Tibshirani, R. J. (1993). *An Introduction to the Bootstrap.* New York: Chapman and Hall.

Gaines, B. R. (1989). An ounce of knowledge is worth a ton of data. *Proceedings of the Sixth International Workshop on Machine Learning* (pp. 156–159). Ithaca, NY: Morgan Kaufmann.

Heckerman, D. (1996). Bayesian networks for knowledge discovery. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, & R. Uthurusamy (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 273–305). Menlo Park, CA: AAAI Press.

Holte, R. C., Acker, L. E., & Porter, B. W. (1989). Concept learning and the problem of small disjuncts. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp. 813–818). Detroit, MI: Morgan Kaufmann.

Huber, P. J. (1994). Languages for statistics and data analysis. In P. Dirschedl & R. Ostermann (Eds.), *Computational Statistics.* Heidelberg: Physica-Verlag.

Jensen, D., & Schmill, M. (1997). Adjusting for multiple comparisons in decision tree pruning. *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining* (pp. 195–198). Newport Beach, CA: AAAI Press.

Kass, G. V. (1980). An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, *29*, 119–127.

Kearns, M. J., & Vazirani, U. V. (1994). *An Introduction to Computational Learning Theory.* Cambridge, MA: MIT Press.

Klockars, A. J., & Sax, G. (1986). *Multiple Comparisons.* Beverly Hills, CA: Sage.

Lawrence, S., Giles, C. L., & Tsoi, A. C. (1997). Lessons in neural network training: Overfitting may be harder than expected. *Proceedings of the Fourteenth National Conference on Artificial Intelligence* (pp. 540–545). Providence, RI: AAAI Press.

MacKay, D. (1992). Bayesian interpolation. *Neural Computation*, *4*, 415–447.

Merz, C. J., Murphy, P. M., & Aha, D. W. (1997). UCI repository of machine learning databases. Department of Information and Computer Science, University of California at Irvine, Irvine, CA.

Michalski, R. S. (1983). A theory and methodology of inductive learning. *Artificial Intelligence*, *20*, 111–161.

Miller, Jr., R. G. (1981). *Simultaneous Statistical Inference* (2nd ed.). New York: Springer-Verlag.

Moody, J. E. (1992). The effective number of parameters: An analysis of generalization and regularization in learning systems. In J. E. Moody, S. J. Hanson, & R. P. Lippmann (Eds.), *Advances in Neural Information Processing Systems 4* (pp. 847–854). San Mateo, CA: Morgan Kaufmann.

Murphy, P., & Pazzani, M. (1994). Exploring the decision forest. *Journal of Artificial Intelligence Research*, *1*, 257–275.

Ng, A. Y. (1997). Preventing "overfitting" of cross-validation data. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 245–253). Nashville, TN: Morgan Kaufmann.

Pagallo, G., & Haussler, D. (1990). Boolean feature discovery in empirical learning. *Machine Learning*, *3*, 71–99.

Pearl, J. (1978). On the connection between the complexity and credibility of inferred models. *International Journal of General Systems*, *4*, 255–264.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (1992). *Numerical Recipes in C* (2nd ed.). Cambridge, UK: Cambridge University Press.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.

Quinlan, J. R., & Cameron-Jones, R. M. (1995). Oversearching and layered search in empirical learning. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 1019–1024). Montréal, Canada: Morgan Kaufmann.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*, 465–471.

Schaffer, C. (1993). Overfitting avoidance as bias. *Machine Learning*, *10*, 153–178.

Schuurmans, D., Ungar, L. H., & Foster, D. P. (1997). Characterizing the generalization performance of model selection strategies. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 340–348). Nashville, TN: Morgan Kaufmann.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 461–464.

St. Amant, R., & Cohen, P. R. (1997). Building an EDA assistant: A progress report. *Preliminary Papers of the Sixth International Workshop on Artificial Intelligence and Statistics* (pp. 501–512). Ft. Lauderdale, FL: Society for Artificial Intelligence and Statistics.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society B*, *36*, 111–147.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory.* New York: Springer-Verlag.

Wallace, C. S., & Boulton, D. M. (1968). An information measure for classification. *Computer Journal*, *11*, 185–194.

Webb, G. I. (1996). Further experimental evidence against the utility of Occam's razor. *Journal of Artificial Intelligence Research*, *4*, 397–417.

Westfall, P. H., & Wolfinger, R. D. (1997). Multiple tests with discrete distributions. *American Statistician*, *51*, 3–8.